2 mirp

X-644-74-3

PREPRINT

FORTRAN IV PROGRAMS FOR SUMMARIZATION AND ANALYSIS OF FRACTURE TRACE AND LINEAMENT PATTERNS

(NASA-TM-X-70596) FORTRAN 4 PROGRAMS FOR SUMMARIZATION AND ANALYSIS OF FRACTURE TRACE AND LINEAMENT PATTERNS (NASA) CSCL 20K

N74-18524

Unclas
G3/32 30850

JANUARY 1974

GSFC

GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND

For information concerning availability
of this document contact:

Technical Information Division, Code 250
Goddard Space Flight Center
Greenbelt, Maryland 20771

(Telephone 301-982-4488)

X-644-74-3

FORTRAN IV PROGRAMS FOR SUMMARIZATION AND ANALYSIS OF FRACTURE TRACE AND LINEAMENT PATTERNS

Melvin H. Podwysocki, Author
Geology Department
University of Maryland
College Park, Maryland 20742
NASA Grant #NGR-21-002-368

and

Paul D. Lowman, Jr.
Project Representative
Planetology Branch
Laboratory for Space Physics
Goddard Space Flight Center
Greenbelt, Maryland 20771

January 1974

GODDARD SPACE FLIGHT CENTER
Greenbelt, Maryland

FORTRAN IV PROGRAMS FOR SUMMARIZATION AND ANALYSIS OF FRACTURE TRACE AND LINEAMENT PATTERNS

ABSTRACT

Systematic and detailed analysis of lineament and fracture trace patterns has long been neglected because of the large number of observations involved in such an analysis. Three FORTRAN IV programs were written to facilitate this manipulation. TRANSFORM converts the initial fracture map data into a format compatible with AZMAP, whose options allow repetitive manipulation of the data for optimization of the analysis. ROSE creates rose diagrams of the fracture patterns suitable for map overlays and tectonic interpretation. Examples are given and further analysis techniques using output from these programs are discussed.

CONTENTS

PRECEDING PAGE BLANK NOT FILMED

ILLUSTRATIONS

FORTRAN IV PROGRAMS FOR SUMMARIZATION AND ANALYSIS OF FRACTURE TRACE AND LINEAMENT PATTERNS

INTRODUCTION

Photogeologic fracture traces and lineaments[1] or other closely related synonyms (micro- and macrofractures[2], megajoints[3], lineaments[4], etc.) have enjoyed a resurgence of interest in the last several years. They have been noted on both large and small scaled imagery of the Earth and surrounding planets[5], and in places on Earth, can in many cases be seen through considerable overburden[6]. Their occurrences and associations with the local and regional tectonic stress patterns are often enigmatic.

One of the major problems of fracture analysis is in handling the large number of observations. Not only does this problem render repetitive manipulation impractical, but also makes the varying of classification parameters to achieve an efficient and sufficient classification cumbersome. As an example, if one wishes to analyze the data for local perturbations in a fracture pattern, the smallest grid size (summation area) that would achieve statistical stability might be desired. Relatively small areas (200 square miles) may contain several thousand fractures. This paper will present a series of computer algorithms which will greatly facilitate fracture (fracture trace)

analysis and will present some results of one particular study. It should be emphasized that the same techniques may be applied to lineament analysis, bearing in mind the changes in scale and their possible significance.

FORTRAN IV PROGRAMS

Purpose

Three programs were written in FORTRAN IV. Their major purpose is to process and summarize in various fashions the large number of primary observations of fracture trace orientation, length and position, while varying the classification parameters so that further statistical or structural analysis techniques might be applied. Input to this group of programs consists of fracture pattern maps plotted on a suitable base, while output consists of frequency-azimuth histograms and/or rose diagrams for the mapped area. Small areas (subsets) of the original mapped areas may be examined by a gridding technique and additional statistical parameters generated.

TRANSFORM Program

TRANSFORM performs the initial data treatment, converting the fracture traces on the base map into a format acceptable for the AZMAP program. A Cartesian coordinate system is established with its origin in the upper left-hand corner of the map. The X-axis is latitudinal and positive to the right

while the Y-axis is meridional and positive downward. North is assumed at the top of the map. Due to the convergence of meridional lines, there will be some discrepancy between true north and the Y-axis for any given point on the map, however, this variation is limited to a total of 2 degrees on a 1:250000 Mercator plot map at 45 degrees latitude.

With the establishment of this coordinate system, the beginning and end of each fracture may now be indexed. Digitizing equipment, which automatically records coordinates either on standard Hollerith computer cards or magnetic tape, is used to reduce the fracture map to a form usable by the program. The digitizer "bull's eye" is set at the origin and a scale is established according to the requirements of the individual machine. Once this is done, the beginning and end points of each fracture may be referenced by placing the "bull's eye" on the beginning (greater Y value) point, recording the value as per the digitizer format on Hollerith cards or magnetic tape, followed by the end point (smaller Y value), repeating the recording procedure. In order to allow the operator to digitize the fractures in a logical manner, it is suggested that each fracture be given an inventory number prior to digitization, beginning with one and consecutively numbered. Most digitizers will generate consecutive numbers on a display panel, thus allowing the operator to constantly check if any fractures have been missed in the digitizing procedure.

The program treats each fracture as a vector in map space and generates parameters used in classification techniques of the AZMAP program. This procedure is only required once for each digitized map, and subsequent repetitive treatments are done in AZMAP and ROSE. The Appendix contains a program listing, abstract, and detailed instructions for data processing. Less than 30 seconds were required to process 1500 vectors on an IBM 360/67 computer.

AZMAP Program

AZMAP uses data generated by the TRANSFORM program. In a mechanical sense, the program places an orthogonal grid over the map parallel to the designated X- and Y- map axes. The operator selects the summation area size (cell) over which the fractures will be summarized by specifying the X- and Y-axis grid cell size. The computer then scans all the fractures, determining whether each fracture falls within the grid cell, incrementing the cell by operator-supplied values in both the X and Y directions, until the total designated area has been covered. By incrementing at submultiples of the cell size a sliding average technique may be employed.

Either of two summary techniques may be used. Subroutine MID counts the whole fracture as falling within the cell if its midpoint falls within that cell, while Subroutine PART considers only that portion of its length which lies

within the cell. The choice of either subroutine depends upon grid cell size, size of the linear features mapped, and goals of the operator or experiment.

The operator also selects the number of azimuth classes into which the data will be summarized. Up to 90 classes may be specified, their values being incremented from 270 through 0 to 90 degrees. Thus, if a fracture trace lies within a specific cell, it is then added to the appropriate azimuth class within that cell according to the subroutine selected, and a histogram is plotted.

Data is summarized both as density (total length of fractures within each cell) and frequency (number of fractures within a cell). Depending upon the technique used, either the whole length (MID) or the portion (PART) is summed for density while a whole unit is added to frequency in either case. Therefore, if a fracture trace extends into or through three cells, frequency will be incremented by one in all three cells if subroutine PART is used.

A Chi Square (χ^2) test is performed on the resultant frequency-azimuth histogram for each grid cell if so desired, testing the distribution for randomness (a rectangular distribution, where all classes have an equal chance of occurring)[7]. Care should be exercised in the interpretation of the χ^2 test, because the lower limit of reliability is reached when the expected frequency for each azimuth class is less than 2 units where:

$$\text{Expected Frequency} = \frac{\text{Total number of units}}{\text{Number of azimuth classes}}$$

Caution is also advised for the density χ^2 test because the test results will be dependent on the scale of the units chosen in this classification.

Figure 1 contains an example of the line printer output for one grid cell. In addition, punched card or magnetic tape output may be generated for use either in the ROSE program or additional statistical programs that will be described but not documented in this paper. The Appendix contains a source listing, abstract, and detailed instructions on the program's use. Timing considerations are dependent upon the summary technique. MID is more efficient and requires less than 60 seconds for summary of 1500 fracture traces into 49 grid cells, while PART requires about 90 seconds on an IBM 360/67 computer.

Although AZMAP requires a relatively large computer, the program may be modified to fit on smaller computers by reducing the size of the arrays designated for storage of the fracture data (dimensioned at 2000) without a resultant decrease in program efficiency. If time is of no objection, then the above mentioned arrays may be removed and with a little rewriting, the fracture data could be stored on tape and repetitively scanned and rewound for each grid cell summarized.

```
          TEXAS STUDY AREA, 1/16TH AREAS, 1/2 CELL INCREMENT

          EACH GRID CELL IS  240 MM. (  3.576 MILE(S)) BY  260 MM. (  3.874 MILE(S))

       PROGRAM USES SUBROUTINE MID; CONSIDERS WHOLE VECTOR AS BEING WITHIN CELL IF ITS MIDPOINT FALLS IN THE  CELL

GRID CELL NUMBER: ROW    4, COLUMN    1 (    0 <X<  240;  390 <Y<  650)

 AZIMUTHS    CLASS LENGTH (IN MILE(S))                               NUMERICAL  FREQUENCY
 --------    -------------------------                               ------------------
270.0-280.0       0.94    >XXXXXX                                           2       >**
280.0-290.0       0.31    >XX                                               1       >*
290.0-300.0       0.98    >XXXXXX                                           4       >****
300.0-310.0       2.91    >XXXXXXXXXXXXXXXXXXX                              7       >*******
310.0-320.0       2.59    >XXXXXXXXXXXXXXXXX                                9       >*********
320.0-330.0       0.89    >XXXXX                                            3       >***
330.0-340.0       1.48    >XXXXXXXXX                                        5       >*****
340.0-350.0       1.27    >XXXXXXXX                                         3       >***
350.0-360.0       0.52    >XXX                                              2       >**
  0.0- 10.0       1.50    >XXXXXXXXXX                                       4       >****
 10.0- 20.0       1.01    >XXXXXX                                           4       >****
 20.0- 30.0       2.46    >XXXXXXXXXXXXXXXX                                 9       >*********
 30.0- 40.0       3.84    >XXXXXXXXXXXXXXXXXXXXXXXXX                       11       >***********
 40.0- 50.0       2.32    >XXXXXXXXXXXXXXX                                  9       >*********
 50.0- 60.0       1.98    >XXXXXXXXXXXXX                                    6       >******
 60.0- 70.0       0.69    >XXXX                                             3       >***
 70.0- 80.0       0.39    >XX                                               1       >*
 80.0- 90.0       0.47    >XXX                                              2       >**
                -------                                                   ----

    TOTALS       26.54       EACH X =   0.15  MILE(S)                      85               EACH * =   1 UNITS

       RANDOMLY DISTRIBUTED DENSITY DATA PROB. = 0.0                  RANDOMLY DISTRIBUTED FREQ. DATA PROB. = 0.7856E-02
```

Figure 1. Example of line printer output from the AZMAP program. Frequency and density histogram and Chi Square test options have been exercised. Grid cell coordinates are given in millimeters and size is given both in millimeters and the actual map units chosen.

ROSE Program

ROSE uses standard CalComp subroutines and hardware, producing rose diagrams from data generated on punched cards or magnetic tape in AZMAP. Those installations lacking this type of plotter or using other software subroutines will either have to forgo this last program, relying on the frequency-azimuth histograms, or restructure the program for use with their particular system. The program will not accommodate three-dimensional data such as jointing. Other programs are in the literature for stereonet plots[8, 9]. Figure 2 represents a plot generated by this program. Less than 40 seconds were required for processing 49 rose diagrams on an IBM 360/67 computer with an additional 5 minutes for plotting on a CalComp 780 30-inch Drum Plotter. A source listing, abstract, and detailed instructions for use are included in the Appendix.

ANALYSIS APPLICATIONS

Discussion

The following will illustrate several applications of these programs to a fracture analysis problem. A fracture trace analysis was performed on a 225 square mile portion of northwestern Nolan and southwestern Fisher Counties, Texas. Figure 3 is a geologic map of the area, located on the eastern shelf of the Permian Basin. It consists of a low relief (<200 feet) terrain

ROSE DIAGRAMS OF FRACTURE TRACE PATTERNS, NOLAN & FISHER COUNTIES, TEXAS.

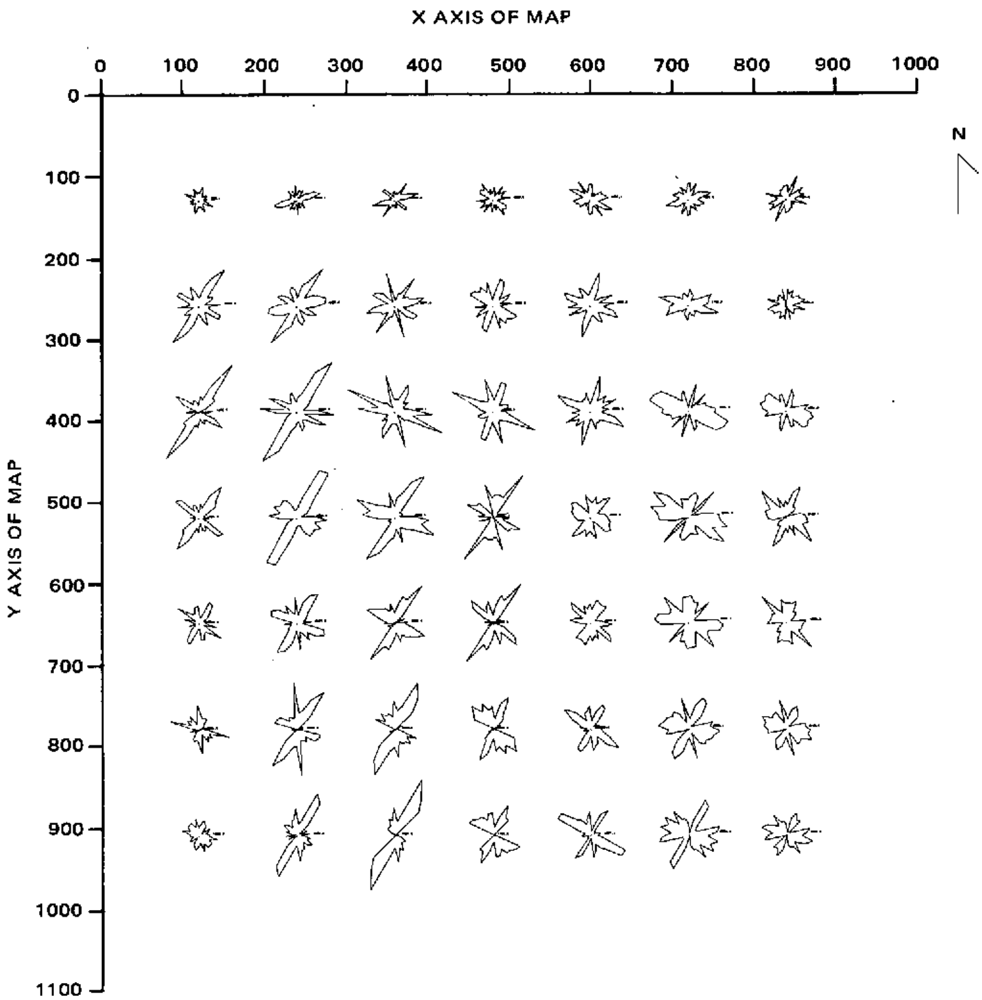

Figure 2. CalComp-generated rose diagram plot of fracture trace patterns.

TEXAS STUDY AREA

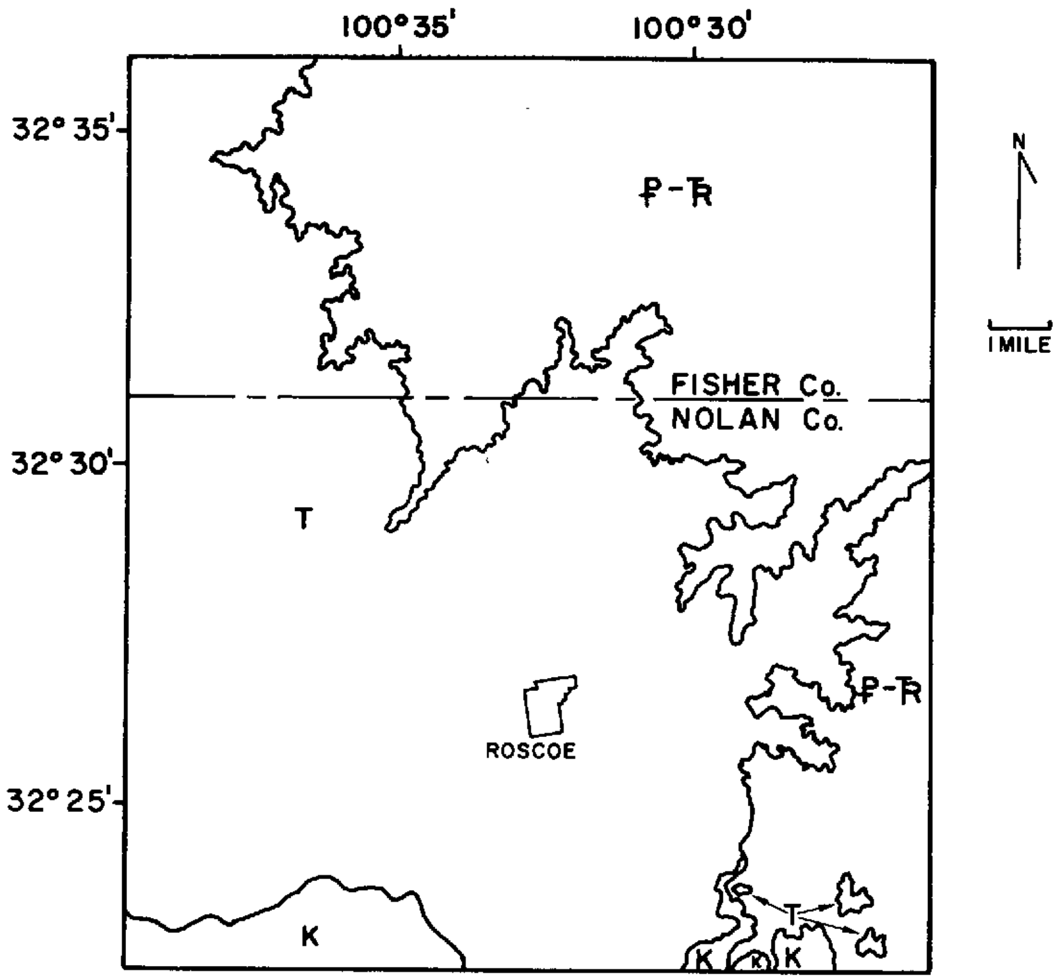

Figure 3. Geologic map of the Texas study area[10, 11].
T - Tertiary Ogallala silts, sands, and gravel overlying predominately Cretaceous limestones of the Trinity and Fredricksburg Groups
K - Cretaceous limestones of the Trinity and Fredricksburg Groups
P-Tr - Permo-Triassic redbeds of gravels, sandstones, siltstones, and shales. Some gypsum horizons.

containing flatlying consolidated (Permian through Cretaceous) and unconsolidated Tertiary sedimentary materials. The fracture traces were mapped on 1:20000 scale aerial photographs in flightlines and then transferred to 1:24000 scale U.S. Geological Survey topographic maps by standard photogrammetric techniques. Figure 4 shows the fracture trace map. The number of observations for this small area illustrates the need for automated data processing. The data were then digitized as discussed earlier and processed by the TRANSFORM program. Output consisting of 1486 fracture traces was subsequently used in AZMAP.

Repetitive manipulations of the fracture data using the various algorithms within AZMAP, while varying the size of the grid cell, allowed an estimate of the grid cell size which provided the most efficient, consistent, and sufficient grid cell size and summarization technique. Based on these findings, a grid cell size of approximately 3.25 x 3.75 miles area was used. To gain further information about the variability of the fracture pattern, summarization was made at 1/2 cell increments. The following is excerpted from Podwysocki[12, 13]. The reader is referred to these publications for more detailed treatment of the data.

Trend surface analysis[14] of the number of fracture traces per grid cell summarized by AZMAP indicates that the third order surface shows fewer fracture traces over the Cretaceous-Tertiary rocks whereas greater

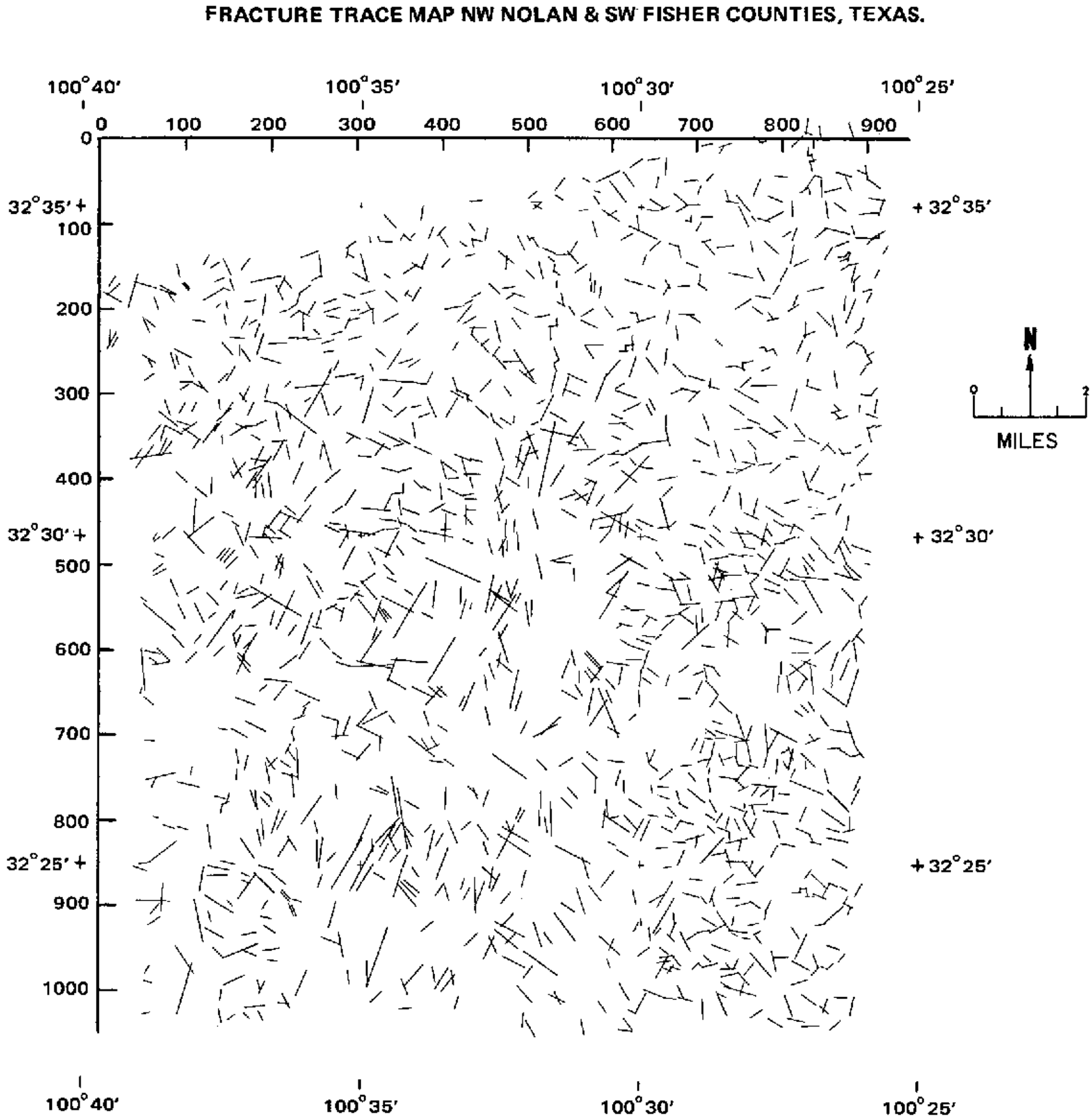

Figure 4. Fracture trace map of portions of Nolan and Fisher Counties, Texas. The upper and left axes correspond to the top and left margins of the rose diagram plot in Figure 2 and maps of Figures 3 and 5.

frequencies are encountered over the Permo-Triassic clastic rocks. This may be interpreted as indicating 1) the older rocks are more fractured, 2) the thin Tertiary mantle (usually less than six feet) has acted as a masking effect, 3) the rock types respond differently to differences in mechanical properties, or 4) two stress fields may have operated on the older rocks while only one affected the younger. Higher-order trends tend to align themselves with the direction of flightlines, a bias most likely attributable to changes in operator accuity. This suggests that photographs should not be mapped in flightlines, but that they should be randomly selected to try to eliminate or distribute this bias evenly during the project.

Cluster analysis, a multivariate analysis technique used to classify samples on the basis of their similarity or variability, is becoming widely used[15,16,17]. A cluster analysis program using covariance measures written by Rubin and Friedman[18] was applied to the frequency-azimuth histograms generated by AZMAP in order to classify areas of like-behaving fracture patterns. Each grid cell was treated as a sample consisting of 18 variables (18 azimuth classes of 10 degrees each). Optimum grouping was determined by plotting the program-generated optimization measure ($\log|T|/|W|$) against the number of groups, which was varied from two through twelve. The greatest rate of increase in the measure was noted between the two and three group levels, with additional minor inflections occurring at higher grouping levels.

Based on the three-group classification, the bedrock geology of the area could be mapped (Figure 5).

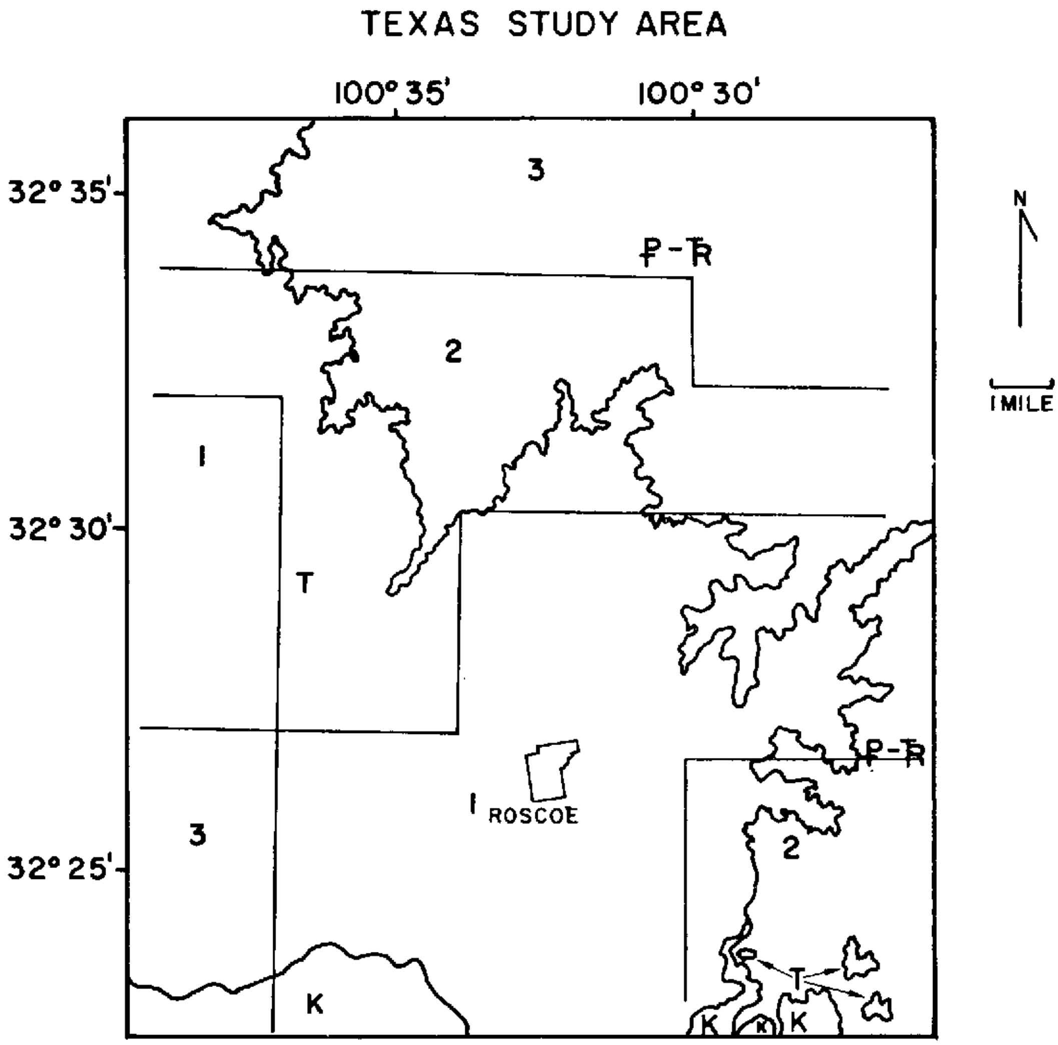

Figure 5. Results of a three group cluster analysis classification of rose diagram patterns observed in Figure 2 based on covariance measures. Group 1 - mainly Cretaceous and Tertiary rocks; group 2 - transition between groups 1 and 3; group 3 -mainly Permian and Triassic rocks.

Conclusions

These programs are an attempt to facilitate easier handling of fracture trace or lineament data in order to provide a basis for the application of rigorous statistical treatment. Users with a knowledge of FORTRAN can, in addition, modify the programs to suit their individual needs. The rose diagram program also facilitates the presentation of the fracture data in a suitable form for qualitative interpretation of the spatial relationships between areas for tectonic analysis. Additional numerical techniques applicable to fracture analysis have been suggested.

ACKNOWLEDGMENTS

The author is grateful to Dr. M. E. Bell of the Earth and Mineral Sciences Experiment Station, College of Earth and Mineral Sciences, The Pennsylvania State University, for the computer support provided during the initial writing of these programs and to Richard H. Ford III, Coal Research Section, The Pennsylvania State University for his assistance during the early stages of program writing. Thanks are also extended to Messrs. S. W. Hoenig, J. V. McKenna, G. T. Masaki and D. Thompson of the Computer Sciences Corporation for assistance in preparing the final versions of these programs and to the NASA Goddard Space Flight Center for use of computer time. Portions of this research were supported by NASA Grant # NGR-21-002-368

and a Chevron Oil Company Fellowship granted through the Geosciences Department, The Pennsylvania State University.

REFERENCES

1. L. H. Lattman. "Techniques of Mapping Geologic Fracture Traces and Lineaments on Aerial Photographs." Photogrammetric Engineering, 24(4), 1958. pp. 568-576.

2. P. H. Blanchet. "Development of Fracture Analysis as an Exploration Method." Bulletin of the American Association of Petroleum Geologists, 41(8), 1957. pp. 1748-1759.

3. I. G. Gol'braikh, V. V. Zabaluyev, and G. R. Mirkin. "Tectonic Analysis of Megajointing; a Promising Method of Investigating Covered Territories." International Geological Review, 8(9), 1968. pp. 1009-1016.

4. S. S. Shul'ts. "Planetary Fractures and Tectonic Deformations." Geotectonics, (4), Academy of Sciences, U.S.S.R., 1971. pp. 203-207.

5. G. N. Katterfel'd and G. V. Charushin. "Global Fracturing on the Earth and Other Planets." Geotectonics, (6), Academy of Sciences, U.S.S.R., 1970. pp. 333-337.

6. F. J. Wobber. "Fracture Traces in Illinois." Photogrammetric Engineering, 33(5), 1967. pp. 499-506.

7. L. H. Lattman and A. V. Segovia. "Analysis of Fracture Trace Patterns of Adak and Kagalaska Islands, Alaska." Bulletin of the American Association of Petroleum Geologists, 45(2), 1961. pp. 249-251.

8. V. T. Loudon. "Computer Analysis of Orientation Data in Structural Geology." Technical Report 13, Office of Naval Research Task 389-135, 1964. 129 pages.

9. H. B. Spencer and P. S. Clabaugh. "Computer Program for Fabric Diagrams." American Journal of Science, 265, 1967. pp. 166-172.

10. A. M. Lloyd and W. C. Thompson. "Areal Map Showing Outcrops on the East Side of the Permian Basin." Texas Bureau of Economic Geology, 1929.

11. W. R. Vance and E. D. Johnson. "Geologic Map of Fisher County, Texas." Texas Bureau of Economic Geology, 1929.

12. M. H. Podwysocki. "Computer Applications of Fracture Trace Analysis (Abs.)." Abstracts with Programs, Geological Society of America, 5(2), 1973. pp. 207-208.

13. M. H. Podwysocki. "An Analysis of Fracture Trace Patterns in Areas of Flat-lying Sedimentary Rocks for the Detection of Buried Geologic Structure." (GSFC X-document in preparation).

14. M. O'Leary, R. H. Lippert, and O. T. Spitz. "FORTRAN IV and Map Program for Computation and Plotting of Trend Surfaces for Degrees 1 Through 6." Computer Contribution 3, Kansas Geological Survey, 1966. 50 pages.

15. J. M. Parks. "Cluster Analysis Applied to Multivariate Geologic Data." Journal of Geology, 74(5), pt. 2, 1966. pp. 703-715.

16. F. H. C. Marriot. "Practical Problems in a Method of Cluster Analysis." Biometrics, 27, 1971. pp. 501-515.

17. F. Block. "A Multivariate Chemical Classification of Rocks from the Monteregian Petrographic Province, Quebec, Canada." Unpublished Ph.D. Thesis, The Penn. St. Univ., Univ. Park, Penn., 1972. 172 pages.

18. J. Rubin and H. P. Friedman. "A Cluster Analysis and Taxonomy System for Grouping and Classifying Data." IBM Corporation, New York, 1967. 221 pages.

APPENDIX

PROGRAM SOURCE LISTINGS

VECTOR TRANSFORM PROGRAM

Preceding page blank

```
C             ****************************************
C                  VECTOR    TRANSFORM    PROGRAM
C             ****************************************
C
C        THE PROGRAM WAS WRITTEN BY MELVIN PODWYSOCKI OF THE GEOSCIENCES
C        DEPT.. THE PENNSYLVANIA STATE UNIVERSITY, APRIL, 1972 FOR USE
C        ON THE IBM 360/67 COMPUTER, AND WAS MODIFIED IN APRIL, 1973
C        FOR USE ON OTHER COMPUTERS HAVING THE EQUIVALENT CORE STORAGE
C        OF 78K BYTES.
C
C      PROGRAM CALCULATES VECTOR LENGTH (VECLEN), AZIMUTH (VECAZM),
C      SLOPE (A), Y INTERCEPT (B) AND X & Y MIDPOINTS (XMID & YMID) FOR
C      EACH VECTOR GIVEN ITS BEGINNING (X1,Y1) AND END (X2,Y2) POINTS
C      IN MAP SPACE.  PROGRAM ASSUMES X IS + TOWARDS THE RIGHT AND Y
C      IS + DOWNWARD. DUE NORTH-SOUTH AND EAST-WEST DATA ARE TREATED AS
C      SPECIAL CASES. CONTROL CARDS ARE READ FROM THE CARD READER WHILE
C      DATA CARDS ARE READ FROM ANY UNIT DECLARED BY 'ITAPE1' ON CONTROL
C      CARD 3. DATA CARDS ARE GENERATED ON ANY UNIT DECLARED BY 'ITAPE2'
C      IN A FORMAT ACCEPTABLE TO "AZMAP" PROGRAM.  COORDINATES MUST BE
C      IN MILLIMETERS, IF NOT, CONTROL CARD 3 ALLOWS A TRANSFORMATION TO
C      BE PERFORMED. PRINTED OUTPUT IS ALSO PRODUCED.
C
C                          (0,0)        X + ---->
C                             ------------------------
C                             |
C                             |
C                             |                          NORTH
C                           Y |
C                             |                            A
C                           + |           ALL VECTORS      |
C                             |           LIE IN THIS      |
C                           | |              AREA
C                           | |
C                           V |
C                             |
C                             |
C
C      ALL NUMERIC INPUT DATA IS RIGHT JUSTIFIED: "I" INDICATES INTEGER
C      FORMAT, "F" INDICATES FLOATING POINT FORMAT,"A" INDICATES CHARAC-
C      TER FORMAT, "#" PRECEEDING NUMBERS INDICATES COLUMNS USED FOR
C      EACH PARAMETER.  TO SPECIFY NONUSE OF AN OPTION, PUNCH 0 OR
C      LEAVE BLANK.
C
C ********CONTROL CARD 1-------TITLE CARD
C     TITLE WILL BE PRINTED AT BEGINNING OF PRINTED OUTPUT (20A4,#1-80)
C ********CONTROL CARD 2-------VARIABLE INPUT FORMAT CARD
C     TO READ X & Y VALUES FROM CARDS.  FORMAT MUST BE ENCLOSED IN
C     PARANTHESES AND BEGIN IN #1. VALUES SHOULD BE READ IN THE FOLLOW-
C     ING ORDER: X1,Y1,X2,Y2 (20A4,#1-80)
C ********CONTROL CARD 3-------OPTIONS CARD
C     ITAPE1=LOGICAL UNIT FOR READING DATA CARDS (I2,#1-2)
C     ITAPE2=LOGICAL UNIT FOR WRITING OUTPUT USEABLE BY "AZMAP" (I2,
C        #3-4)
C     NPRINT--PRINT COMMAND FOR OUTPUT (I1,1 IN #5)
C     NPUNCH--PUNCH COMMAND FOR OUTPUT ACCORDING TO FORMAT FOR "AZMAP"
C        PROGRAM ON ANY UNIT DECLARED BY 'ITAPE2' (I1,1 IN #6)
C     NTRAN--TRANSFORMATION COMMAND TO CONVERT UNITS OF INPUT DATA TO
C        MILLIMETERS  (I1,1 IN #7)
C     NOTE: THE FOLLOWING PARAMETER IS USED ONLY IF 'NTRAN' IS PUNCHED
C        1. UNITS WILL BE CONVERTED TO MILLIMETERS FOR USE IN "AZMAP".
```

```
C         CONVERTED COORDINATE VALUE   = ORIGINAL VALUE * 'CONV'
C      CONV=TRANSFORMATION FACTOR TO CONVERT UNITS ON INPUT DATA CARDS
C         TO MILLIMETERS   (F6.2,#8-13)
C *******DATA CARDS------
C      READ FROM ANY UNIT DECLARED BY 'ITAPE1'. END OF DATA CARDS IS
C         SIGNIFIED WHEN THE PROGRAM ENCOUNTERS A CARD WHOSE VALUES OF
C         X1,Y1,X2 & Y2 ARE ALL 0.0
C
      DIMENSION FMTRD(20),TITLE(20)
      DATA IREAD/5/,IPRINT/6/
C
C      READ CONTROL CARDS
C
      READ(IREAD,5)  (TITLE(L),L=1,20)
      READ(IREAD,5)  (FMTRD(L),L=1,20)
      READ(IREAD,7)  ITAPE1,ITAPE2,NPRINT,NPUNCH,NTRAN,CONV
      WRITE(IPRINT,9)  (TITLE(L),L=1,20),(FMTRD(L),L=1,20)
      IF(NTRAN.EQ.1) WRITE(IPRINT,10) CONV
      IF(NPRINT.EQ.1) WRITE(IPRINT,11)
C
C      READ X & Y COORDINATES FROM UNIT 'ITAPE1'
C
      NUM=0
   12 READ(ITAPE1,FMTRD)  X1,Y1,X2,Y2
      IF(X1.EQ.0..AND.Y1.EQ.0..AND.X2.EQ.0..AND.Y2.EQ.0.) GO TO 100
      NUM=NUM+1
C
C      CONVERT DATA TO MILLIMETERS IF NECESSARY
C
      IF(NTRAN) 20,20,15
   15 X1=X1*CONV
      Y1=Y1*CONV
      X2=X2*CONV
      Y2=Y2*CONV
C
C      ORDER BEGINNING AND END COORDINATES FOR "AZMAP" PROGRAM
C
   20 IF(Y1.EQ.Y2.AND.X1.LT.X2) GO TO 22
      IF(Y1-Y2) 21,23,23
   21 Z2=Y2
      Y2=Y1
      Y1=Z2
   22 Z1=X2
      X2=X1
      X1=Z1
C
C      CALCULATE VECLEN
C
   23 VECLEN=SQRT(((X2-X1)**2)+((Y2-Y1)**2))
C
C      CALCULATE SLOPE 'A'
C
      IF(X2-X1) 24,25,24
   24 A=(Y2-Y1)/(X2-X1)
      GO TO 30
   25 A=-573.0
      VECAZM=0.0
      B=500000.
      GO TO 50
C
C      CALCULATE VECAZM
```

```
C
   30 IF(A) 35,33,33
   33 VECAZM=270.0 + ((ATAN(A)*180.)/3.14159)
      GO TO 40
   35 VECAZM=90. - ((ATAN(ABS(A))*180.)/3.14159)
C
C     CALCULATE Y INTERCEPT 'B'
C
   40 B=(((Y1-Y2)*X2)/(X2-X1))+Y2
C
C     CALCULATE MIDPOINT OF VECTOR
C
   50 XMID=(X2+X1)/2.0
      YMID=(Y2+Y1)/2.0
C
C     OUTPUT
C
      IF(NPRINT) 75,75,62
   62 WRITE(IPRINT,65)  X1,Y1,X2,Y2,VECLEN,VECAZM,A,B,XMID,YMID
   75 IF(NPUNCH) 12,12,77
   77 WRITE(ITAPE2,80)  X1,Y1,X2,Y2,VECLEN,VECAZM,A,B,XMID,YMID
      GO TO 12
  100 WRITE(IPRINT,105) NUM
    5 FORMAT (20A4)
    7 FORMAT (2I2,3I1,F6.2)
    9 FORMAT (1H1,20A4/,1H0,35HVARIABLE INPUT FORMAT FOR X & Y IS ,20A4)
   10 FORMAT(1H0,44HALL VALUES WILL BE MULTIPLIED BY A FACTOR OF,F12.5)
   11 FORMAT (1H0,4X,2HX1,11X,2HY1,11X,2HX2,11X,2HY2,9X,6HVECLEN,5X,6HVE
     1CAZM,9X,1HA,13X,1HB,12X,4HXMID,9X,4HYMID//)
   65 FORMAT(1H ,5(F7.1,6X),F5.1,6X,F8.3,6X,F9.1,2(6X,F7.1))
   80 FORMAT  (6F6.1,F9.4,F10.1,2F7.1)
  105 FORMAT (1H0,20HNUMBER OF VECTORS = ,I10)
      STOP
      END
```

AZMAP PROGRAM

```
C                    *****************************                      AZMP0010
C                         AZMAP       PROGRAM                            AZMP0020
C                    *****************************                      AZMP0030
C                                                                        AZMP0040
C       THE PROGRAM WAS WRITTEN BY MELVIN PODWYSOCKI OF THE GEOSCIENCESAZMP0050
C       DEPT., THE PENNSYLVANIA STATE UNIVERSITY, APRIL, 1972 FOR THE   AZMP0060
C       IBM 360/67 COMPUTER, AND WAS MODIFIED IN APRIL, 1973, FOR USE   AZMP0070
C       ON OTHER COMPUTERS HAVING THE EQUIVALENT OF 160K BYTES STORAGE.AZMP0080
C                                                                        AZMP0090
C     PROGRAM SUMMARIZES FREQUENCY DISTRIBUTIONS  OF VECTOR DATA IN     AZMP0100
C     VARIABLE MAP GRID AND AZIMUTH CLASS SIZES. PROGRAM ALLOWS UP TO   AZMP0110
C     90 AZIMUTH CLASSES, FROM 270 THRU 0 TO 90 DEGREES, SUMMARIZING    AZMP0120
C     DATA AS TOTAL LENGTH OF VECTORS/AZIMUTH CLASS (DENSITY) OR NUM-   AZMP0130
C     BER OF VECTORS/AZIMUTH CLASS (FREQUENCY).  UP TO 2000 VECTORS     AZMP0140
C     MAY BE USED.  X AXIS IS + TO RIGHT & Y AXIS IS + DOWNWARD. DATA   AZMP0150
C     ARE READ FROM CARDS GENERATED BY VECTOR TRANSFORM PROGRAM.        AZMP0160
C     CONTROL & TITLE CARDS ARE READ FROM CARD READER WHILE DATA CARDS AZMP0170
C     MAY BE READ FROM ANY UNIT DECLARED BY 'ITAPE2' ON CONTROL CARD 1.AZMP0180
C                                                                        AZMP0190
C     ALL NUMERIC INPUT DATA IS RIGHT JUSTIFIED; "I" INDICATES INTEGER AZMP0200
C     FORMAT, "F" INDICATES FLOATING POINT FORMAT, "A" INDICATES CHA-   AZMP0210
C     RACTER FORMAT, "#" PRECEEDING NUMBERS INDICATES COLUMNS USED FOR AZMP0220
C     EACH PARAMETER. TO SPECIFY NONUSE OF AN OPTION, PUNCH 0 OR LEAVE AZMP0230
C     BLANK.                                                             AZMP0240
C                                                                        AZMP0250
C ********CONTROL CARD 1-------OPTIONS CARD                               AZMP0260
C    XINC=INCREMENT OF X-AXIS TRAVERSE IN MILLIMETERS (I4,#1-4)          AZMP0270
C    YINC=INCREMENT OF Y-AXIS TRAVERSE IN MILLIMETERS (I4,#5-8)          AZMP0280
C    XSTART=STARTING POINT FOR X-AXIS TRAVERSE IN MILLIMETERS(I4,9-12) AZMP0290
C    YSTART=STARTING POINT FOR Y-AXIS TRAVERSE IN MILLIMETERS(I4,#13-16AZMP0300
C    XSTOP=END OF X-AXIS TRAVERSE IN MILLIMETERS (I4,#17-20)            AZMP0310
C    YSTOP=END OF Y-AXIS TRAVERSE IN MILLIMETERS (I4,#21-24)            AZMP0320
C       NOTE: PROGRAM SUCCESSIVELY SCANS DATA IN MAP GRID CELLS 'XCELL'AZMP0330
C       BY 'YCELL' IN SIZE, INCREMENTING BY 'XINC' UNTIL 'XMAX' >       AZMP0340
C       'XSTOP', WHEN 'YINC' IS INCREMENTED.  PROGRAM TERMINATES WHEN   AZMP0350
C       'YMAX' > 'YSTOP'. NONE OF THE ABOVE 6 VALUES CAN BE NEGATIVE.   AZMP0360
C    AZCLAS=AZIMUTH CLASS WIDTH IN DEGREES  (F3.1,#25-27)               AZMP0370
C    AMPSCL=MAPSCALE IN UNITS/SCALE; SEE SCALE PARAM. BELOW (F5.4,#28-  AZMP0380
C       32)                                                              AZMP0390
C    KTYPE--SELECTS SUBROUTINE FOR CLASSIFYING VECTOR DATA (I2,#33-34) AZMP0400
C       PUNCH 1 FOR SUBR. PART.  CONSIDERS ONLY THAT PART OF VECTOR     AZMP0410
C       WHICH LIES WITHIN THAT CELL                                      AZMP0420
C       PUNCH -1 FOR SUBR. MID.  CONSIDERS WHOLE VECTOR IN CELL IF ITS AZMP0430
C       MIDPOINT FALLS WITHIN THAT CELL                                  AZMP0440
C    NDHIST--PRINT COMMAND FOR DENSITY HISTOGRAM (I1,1 IN #35)          AZMP0450
C    NFHIST--PRINT COMMAND FOR FREQUENCY HISTOGRAM (I1,1 IN #36)        AZMP0460
C    NPUNCH--PUNCHED CARD OUTPUT FOR EACH GRID CELL AZIMUTH DISTRIBU-   AZMP0470
C       TION IS GENERATED AS PER CONTROL CARD 2A BELOW (I1,1 IN #37)    AZMP0480
C    DHINC=NUMERICAL VALUE OF EACH 'X' INCREMENT (ACCORDING TO SCALE    AZMP0490
C       PARAMETER; SEE BELOW) FOR DENSITY HISTOGRAM (F5.2,#38-42)       AZMP0500
C    NFHINC=NUMERICAL VALUE OF EACH '*' INCREMENT FOR FREQUENCY HISTO-  AZMP0510
C       GRAM (I3,#43-45)                                                 AZMP0520
C    SCALE=SCALE UNITS FOR PRINTOUT (I.E.MILES,MM.,ETC.),(2A4,#46-53)   AZMP0530
C       NOTE: WHEN VECTORS ARE MEASURED ON A 1:24000 SCALE MAP AND OUT-AZMP0540
C       PUT IS DESIRED IN MILES,'AMPSCL'=.0149 (I.E. 1 MM.=.0149 MILES)AZMP0550
C    ITAPE1=LOGICAL UNIT FOR READING DATA CARDS GENERATED BY "TRANS-    AZMP0560
C       FORM" PROGRAM (I2,#54-55)                                        AZMP0570
C    NCHI--COMMAND TO TEST FREQUENCY AND DENSITY DATA FOR RANDOMNESS    AZMP0580
C       IN EACH GRID CELL (I1, 1 IN #60)                                 AZMP0590
C    XCELL=CELL SIZE IN X DIRECTION (I4,#61-64)                          AZMP0600
```

PRECEDING PAGE BLANK NOT FILMED

```
C     YCELL=CELL SIZE IN Y DIRECTION (I4,#65-68)
C     NUM=NUMBER OF VECTORS (AS GIVEN IN TRANSFORM PROGRAM) (I4,#69-72)
C ********CONTROL CARD 2-------TITLE CARD
C     TITLE WILL BE PRINTED AT TOP OF EACH HISTOGRAM (20A4,#1-80)
C ********CONTROL CARD 2A-------VARIABLE OUTPUT FORMAT CARD
C     TO BE USED ONLY IF NPUNCH (CONTROL CARD 1) IS PUNCHED 1
C     PUNCHED OUTPUT WILL CONSIST OF A CELL ROW, COLUMN, X & Y MIDPOINTS
C        OF CELL, AND THE VALUE OF EACH AZIMUTH CLASS SUMMATION AS PER
C        VARIABLE 'ITYPE'. THE FOLLOWING FORMAT IS SUGGESTED:
C                 (2I4,2F7.2,8F7.2/10F7.2)
C        NOTE: THE LAST SET OF VARIABLES MUST CORRESPOND TO THE NUMBER
C        OF AZIMUTH CLASSES (I.E. 180./'AZCLAS' = NUMBER OF CLASSES).
C        IN THE ABOVE EXAMPLE IT'S 18. FOR DENSITY DATA IT SHOULD BE AS
C        ABOVE, BUT IN INTEGER FORMAT, I.E. (.........,8I5/10I5)
C     FMTPCH--OUTPUT FORMAT FOR AZIMUTH CLASS DATA. MUST BE ENCLOSED
C        IN PARANTHESES AND START IN #1. (18A4, #1-72)
C     ITYPE--PUNCH 2 FOR DENSITY DATA (TOTAL LENGTH/AZIMUTH CLASS)
C        PUNCH 1 FOR FREQUENCY DATA (NUMBER OF VECTORS/AZIMUTH CLASS)
C        OUTPUT WILL BE IN UNITS SPECIFIED BY 'AMPSCL' AND 'SCALE'
C        (I1,#77)
C ********DATA CARDS------
C     VECTOR DATA INPUT FROM VECTOR TRANSFORM PROGRAM
C
C
      DIMENSION TITLE(20),SCALE(2),FMTPCH(18),Z3(90)
      COMMON X1(2000),Y1(2000),X2(2000),Y2(2000),VECAZM(2000),VECLEN(200
     10),A(2000),B(2000),XMID(2000),YMID(2000),CLAMIN(90),CLAMAX(90),AZL
     2EN(90),NAZFRQ(90),XMIN,YMIN,XMAX,YMAX,FLAG,VLEN,INUM
      INTEGER XSTOP,YSTOP,XINC,YINC,XSTART,YSTART,XMIN,YMIN,XMAX,YMAX,FL
     1AG,XCELL,YCELL
      DATA IXS/1HX/,ISTARS/1H*/,IREAD/5/,IPRINT/6/,IPUNCH/7/
C
C     READ CONTROL INPUT INFORMATION AND TITLE CARD
C
      READ(IREAD,5) XINC,YINC,XSTART,YSTART,XSTOP,YSTOP,AZCLAS,AMPSCL,
     1KTYPE,NDHIST,NFHIST,NPUNCH,DHINC,NFHINC,(SCALE(L),L=1,2),ITAPE1,
     2NCHI,XCELL,YCELL,NUM
      READ(IREAD,6) (TITLE(L),L=1,20)
      IF(NPUNCH.EQ.1) READ(IREAD,7) (FMTPCH(L),L=1,18),ITYPE
      Z1=XCELL*AMPSCL
      Z2=YCELL*AMPSCL
      IF(KTYPE) 19,9,19
    9 WRITE(IPRINT,10)
      STOP
C
C     READ DATA GENERATED BY VECTOR TRANSFORM PROGRAM
C
   19 DO 25 I=1,NUM
      READ(ITAPE1,20) X1(I),Y1(I),X2(I),Y2(I),VECLEN(I),VECAZM(I),A(I),
     1B(I),XMID(I),YMID(I)
   25 CONTINUE
C
C     GENERATE AZIMUTH CLASSES
C
   30 CLAMIN(1)=270.0
      I=1
      NCLASS=1
      CLAMAX(I)=CLAMIN(I)+AZCLAS
   40 IF(CLAMAX(I).GT.270.0.AND.CLAMAX(I).LE.360.0) GO TO 41
      IF(CLAMAX(I)-90.0) 41,60,60
   41 I=I+1
```

```
      NCLASS=I
      CLAMIN(I)=CLAMAX(I-1)
      CLAMAX(I)=CLAMIN(I)+AZCLAS
      IF(CLAMAX(I)-360.0) 40,50,50
   50 NCLASS=I+1
      I=NCLASS
      CLAMIN(I)=CLAMAX(I-1)-360.0
      CLAMAX(I)=CLAMIN(I)+AZCLAS
      GO TO 40
C
C     SCAN & SUMMARIZE DATA FOR EACH GRID CELL
C
   60 DO 300 YMIN=YSTART,YSTOP,YINC
      YMAX=YMIN+YCELL
      IF(YMAX-YSTOP) 62,62,400
   62 DO 300 XMIN=XSTART,XSTOP,XINC
      XMAX=XMIN+XCELL
      IF(XMAX-XSTOP) 64,64,300
   64 DO 65 L=1,90
      AZLEN(L)=0.
   65 NAZFRQ(L)=0
      WVLEN=0.
      NEWFRQ=0
      DO 170 I=1,NUM
      INUM=I
      FLAG=0
      IF(KTYPE)68,9,66
   66 CALL PART
      GO TO 69
   68 CALL MID
   69 IF(FLAG)70,75,70
C
C     TEST FOR E - W DATA
C
   70 IF (VECAZM(I)-270.) 76,71,76
   71 IF(FLAG)72,170,73
   72 WVLEN=WVLEN+VECLEN(I)
      GO TO 74
   73 WVLEN=WVLEN+VLEN
   74 NEWFRQ=NEWFRQ+1
C
C     ADD RESULTS TO APPROPRIATE AZIMUTH CLASS
C
   75 IF(I-NUM) 170,78,78
   76 IF(I-NUM) 82,78,78
   78 IF(AZLEN(1)-AZLEN(NCLASS)) 79,80,80
   79 AZLEN(NCLASS)=AZLEN(NCLASS)+WVLEN
      NAZFRQ(NCLASS)=NAZFRQ(NCLASS)+NEWFRQ
      GO TO 81
   80 AZLEN(1)=AZLEN(1)+WVLEN
      NAZFRQ(1)=NAZFRQ(1)+NEWFRQ
   81 IF(VECAZM(I).NE.270..AND.FLAG.NE.0.AND.I.EQ.NUM) GO TO 82
      GO TO 170
   82 DO 85 J=1,NCLASS
      IF (VECAZM(I).GE.CLAMIN(J).AND.VECAZM(I).LT.CLAMAX(J)) GO TO 84
      GO TO 85
   84 NTYPE=J
      GO TO 90
   85 CONTINUE
   90 IF(FLAG)100,170,150
  100 AZLEN(NTYPE)=AZLEN(NTYPE)+VECLEN(I)
```

```
      GO TO 160                                                         AZMP1830
  150 AZLEN(NTYPE)=AZLEN(NTYPE)+VLEN                                    AZMP1840
  160 NAZFRQ(NTYPE)=NAZFRQ(NTYPE)+1                                     AZMP1850
  170 CONTINUE                                                          AZMP1860
C                                                                       AZMP1870
C     OUTPUT                                                            AZMP1880
C                                                                       AZMP1890
      TOTLN=0.                                                          AZMP1900
      NFRQ=0                                                            AZMP1910
      NXERR=0                                                           AZMP1920
      NASTER=0                                                          AZMP1930
      DO 180 N=1,NCLASS                                                 AZMP1940
      TOTLN=TOTLN+AZLEN(N)                                              AZMP1950
      NFRQ=NFRQ+NAZFRQ(N)                                               AZMP1960
  180 CONTINUE                                                          AZMP1970
C                                                                       AZMP1980
C     TEST FOR RANDOMNESS OF AZIMUTH DISTRIBUTIONS/CELL BY CHI SQUARE   AZMP1990
C                                                                       AZMP2000
      IF(NCHI) 184,184,181                                              AZMP2010
  181 CLASS=NCLASS                                                      AZMP2020
      FRQ=NFRQ                                                          AZMP2030
      DENEXP=TOTLN/CLASS                                                AZMP2040
      FRQEXP=FRQ/CLASS                                                  AZMP2050
      DCS=0.                                                            AZMP2060
      FCS=0.                                                            AZMP2070
      DO 182 LCS=1,NCLASS                                               AZMP2080
      DCS=DCS+((AZLEN(LCS)-DENEXP)**2)/DENEXP                           AZMP2090
      FCS=FCS+((NAZFRQ(LCS)-FRQEXP)**2)/FRQEXP                          AZMP2100
  182 CONTINUE                                                          AZMP2110
      NDF=NCLASS-1                                                      AZMP2120
      DCHPRB=PRBCHI(DCS,NDF)                                            AZMP2130
      FCHPRB=PRBCHI(FCS,NDF)                                            AZMP2140
C                                                                       AZMP2150
C     PRINT DATA FOR EACH CELL                                          AZMP2160
C                                                                       AZMP2170
  184 NROW=(YMIN+YINC)/YINC                                             AZMP2180
      NCOL=(XMIN+XINC)/XINC                                             AZMP2190
      GXMID=(XMIN+XMAX)/2.                                              AZMP2200
      GYMID=(YMIN+YMAX)/2.                                              AZMP2210
      WRITE(IPRINT,185) (TITLE(L),L=1,20)                               AZMP2220
      WRITE(IPRINT,186) XCELL,Z1,(SCALE(L),L=1,2),YCELL,Z2,(SCALE(L),L=1AZMP2230
     1,2)                                                               AZMP2240
      IF(KTYPE) 195,9,196                                               AZMP2250
  195 WRITE(IPRINT,188)                                                 AZMP2260
      GO TO 198                                                         AZMP2270
  196 WRITE(IPRINT,187)                                                 AZMP2280
  198 WRITE(IPRINT,190) NROW,NCOL,XMIN,XMAX,YMIN,YMAX                   AZMP2290
      WRITE(IPRINT,200) (SCALE(L),L=1,2)                                AZMP2300
      DO 230 I=1,NCLASS                                                 AZMP2310
      Z3(I)=AZLEN(I)*AMPSCL                                             AZMP2320
      WRITE(IPRINT,205) CLAMIN(I),CLAMAX(I),Z3(I),NAZFRQ(I)             AZMP2330
      IF(NDHIST)220,220,214                                             AZMP2340
  214 NUMX=Z3(I)/DHINC                                                  AZMP2350
      IF(NUMX)217,215,217                                               AZMP2360
  215 WRITE(IPRINT,216)                                                 AZMP2370
      GO TO 220                                                         AZMP2380
  217 IF(NUMX-50)219,219,218                                            AZMP2390
  218 NUMX=50                                                           AZMP2400
      NXERR=1                                                           AZMP2410
  219 WRITE(IPRINT,286) (IXS,IUKA=1,NUMX)                               AZMP2420
  220 IF(NFHIST)230,230,221                                             AZMP2430
```

```
  221 NUMAST=NAZFRQ(I)/NFHINC                                            AZMP2440
      IF(NUMAST)224,222,224                                              AZMP2450
  222 WRITE(IPRINT,223)                                                  AZMP2460
      GO TO 230                                                          AZMP2470
  224 IF(NUMAST-43)226,226,225                                           AZMP2480
  225 NUMAST=43                                                          AZMP2490
      NASTER=1                                                           AZMP2500
  226 WRITE(IPRINT,287) (ISTARS,LIBRAL=1,NUMAST)                         AZMP2510
  230 CONTINUE                                                           AZMP2520
C                                                                        AZMP2530
C     PUNCH CARD OUTPUT AS PER CONTROL CARD 2A                           AZMP2540
C                                                                        AZMP2550
      IF(NPUNCH) 252,252,234                                             AZMP2560
  234 IF(ITYPE-1) 330,238,236                                            AZMP2570
  236 WRITE(IPUNCH,FMTPCH) NROW,NCOL,GXMID,GYMID,(Z3(KQA),KQA=1,NCLASS) AZMP2580
      GO TO 252                                                          AZMP2590
  238 WRITE(IPUNCH,FMTPCH) NROW,NCOL,GXMID,GYMID,(NAZFRQ(KQA),KQA=1,     AZMP2600
     1NCLASS)                                                            AZMP2610
C                                                                        AZMP2620
C     PRINT SUMMARY DATA FOR EACH CELL                                   AZMP2630
C                                                                        AZMP2640
  252 WRITE(IPRINT,240)                                                  AZMP2650
      TLENM=TOTLN*AMPSCL                                                 AZMP2660
      WRITE(IPRINT,250) TLENM,NFRQ                                       AZMP2670
      IF(NDHIST)270,270,260                                              AZMP2680
  260 WRITE(IPRINT,265) DHINC,(SCALE(L),L=1,2)                           AZMP2690
      IF(NXERR.EQ.1) WRITE(IPRINT,290)                                   AZMP2700
  270 IF(NFHIST)296,296,280                                              AZMP2710
  280 WRITE(IPRINT,285) NFHINC                                           AZMP2720
      IF(NASTER.EQ.1) WRITE(IPRINT,295)                                  AZMP2730
  296 IF(NCHI.EQ.1) WRITE(IPRINT,297) DCHPRB,FCHPRB                      AZMP2740
  300 CONTINUE                                                           AZMP2750
    5 FORMAT(6I4,F3.1,F5.4,I2,3I1,F5.2,I3,2A4,I2,4X,I1,3I4)              AZMP2760
    6 FORMAT(20A4)                                                       AZMP2770
    7 FORMAT(18A4,4X,I1)                                                 AZMP2780
   10 FORMAT(1H1,30X,81HNEITHER SUMMARIZATION TECHNIQUE (PART OR MID) WAAZMP2790
     1S SPECIFIED.*****JOB ABORTED*****)                                 AZMP2800
   20 FORMAT(6F6.1,F9.4,F10.1,2F7.1)                                     AZMP2810
  185 FORMAT(1H1,20X,20A4)                                               AZMP2820
  186 FORMAT(1H0,20X, 18HEACH GRID CELL IS ,I4,6H MM. (,F7.3,2A4,5H) BY AZMP2830
     1,I4,6H MM. (,F7.3,2A4,1H))                                         AZMP2840
  187 FORMAT(1H0,20X, 88HPROGRAM USES SUBROUTINE PART; CONSIDERS ONLY THAZMP2850
     1AT PORTION OF EACH VECTOR WITHIN THE CELL)                         AZMP2860
  188 FORMAT(1H0,15X, 107HPROGRAM USES SUBROUTINE MID; CONSIDERS WHOLE VAZMP2870
     1ECTOR AS BEING WITHIN CELL IF ITS MIDPOINT FALLS IN THE  CELL)     AZMP2880
  190 FORMAT(1H0,22HGRID CELL NUMBER: ROW ,I4,9H, COLUMN ,I4,2H (,I4,5H AZMP2890
     1<X< ,I4,2H; ,I4,5H <Y< ,I4,1H))                                    AZMP2900
  200 FORMAT(1H0,/,10H  AZIMUTHS, 4X,16HCLASS LENGTH (IN,2A4,1H),37X,19HAZMP2910
     1NUMERICAL FREQUENCY,/,2X,8(1H-), 4X,25(1H-),37X,19(1H-))           AZMP2920
  205 FORMAT(1H ,F5.1,1H-,F5.1, 2X,F8.2,57X ,I4)                         AZMP2930
  216 FORMAT(1H+,24X,1H>)                                                AZMP2940
  223 FORMAT(1H+,87X,1H>)                                                AZMP2950
  240 FORMAT(1H ,14X,7H-------,58X, 4H----)                              AZMP2960
  250 FORMAT(1H0,5X,6HTOTALS,2X,F8.2,56X,I5)                             AZMP2970
  265 FORMAT(1H+,28X,9HEACH X = ,F7.2,1X,2A4)                            AZMP2980
  285 FORMAT(1H+,95X,9HEACH * = ,I4,6H UNITS)                            AZMP2990
  286 FORMAT(1H+,24X,1H>,50A1)                                           AZMP3000
  287 FORMAT(1H+,87X,1H>,43A1)                                           AZMP3010
  290 FORMAT(1H ,59HONE OR MORE DENSITY HISTOGRAM CLASSES EXCEED LIMITS AZMP3020
     1ALLOWED)                                                           AZMP3030
  295 FORMAT(1H ,61HONE OR MORE FREQUENCY HISTOGRAM CLASSES EXCEED LIMITAZMP3040
```

```
     1S ALLOWED)                                                        AZMP3050
  297 FORMAT(1H0,13X,42HRANDOMLY DISTRIBUTED DENSITY DATA PROB. = ,E10.4AZMP3060
     1,12X,40HRANDOMLY DISTRIBUTED FREQ. DATA PROB. = ,E10.4)           AZMP3070
  315 FORMAT(1H1,30X,71H'ITYPE' FOR PUNCHED OUTPUT NOT IN SPECIFIED RANGAZMP3080
     1E.*****JOB ABORTED*****)                                          AZMP3090
  330 WRITE(IPRINT,315)                                                 AZMP3100
  400 STOP                                                              AZMP3110
      END                                                               AZMP3120
      SUBROUTINE PART                                                   PRT00010
C                                                                       PRT00020
C      SUBROUTINE 'PART' DETERMINES IF A VECTOR FALLS WITHIN A GRID     PRT00030
C      CELL, AND IF SO, THE LENGTH OF THE PORTION WITHIN THE CELL       PRT00040
C                                                                       PRT00050
      COMMON X1(2000),Y1(2000),X2(2000),Y2(2000),VECAZM(2000),VECLEN(200PRT00060
     10),A(2000),B(2000),XMID(2000),YMID(2000),CLAMIN(90),CLAMAX(90),   PRT00070
     2AZLEN(90),NAZFRQ(90),XMIN,YMIN,XMAX,YMAX,FLAG,VLEN,I              PRT00080
      INTEGER YMIN,XMIN,YMAX,XMAX,FLAG                                  PRT00090
      REAL MINLEN                                                       PRT00100
      DATA IREAD/5/,IPRINT/6/,MINLEN/1.0/                               PRT00110
C                                                                       PRT00120
C     DETERMINES IF WHOLE VECTOR IS WITHIN GRID CELL                    PRT00130
C                                                                       PRT00140
C     ANY VECTOR < MINLEN (IN MM.) WILL NOT BE COUNTED                  PRT00150
C                                                                       PRT00160
      IF(X1(I).GE.XMIN.AND.X1(I).LE.XMAX.AND.Y1(I).GE.YMIN.AND.Y1(I).LE.PRT00170
     1YMAX.AND.X2(I).GE.XMIN.AND.X2(I).LE.XMAX.AND.Y2(I).GE.YMIN.AND.Y2(PRT00180
     2I).LE.YMAX) GO TO 1                                               PRT00190
      GO TO 2                                                           PRT00200
    1 FLAG =-1                                                          PRT00210
      RETURN                                                            PRT00220
    2 IF(A(I)) 3,20,3                                                   PRT00230
    3 IF(A(I).LE.-573..AND.X1(I).GE.XMAX) GO TO 422                     PRT00240
      IF(X1(I).EQ.XMAX.AND.A(I).GT.0.) GO TO 20                         PRT00250
      IF(X1(I).GE.XMIN.AND.X1(I).LT.XMAX.AND.Y1(I).GE.YMIN.AND.Y1(I).LT.PRT00260
     1YMAX) GO TO 100                                                   PRT00270
      IF(X2(I).GE.XMIN.AND.X2(I).LT.XMAX.AND.Y2(I).GE.YMIN.AND.Y2(I).LT.PRT00280
     1YMAX) GO TO 200                                                   PRT00290
      GO TO 300                                                         PRT00300
   20 IF(X1(I).GE.XMIN.AND.X1(I).LE.XMAX.AND.Y1(I).GE.YMIN.AND.Y1(I).LT.PRT00310
     1YMAX) GO TO 100                                                   PRT00320
      IF(X2(I).GE.XMIN.AND.X2(I).LE.XMAX.AND.Y2(I).GE.YMIN.AND.Y2(I).LT.PRT00330
     1YMAX) GO TO 200                                                   PRT00340
      GO TO 300                                                         PRT00350
C                                                                       PRT00360
C     CALCULATES VECLEN WHEN VECTOR ORIGIN IS WITHIN GRID CELL          PRT00370
C                                                                       PRT00380
  100 IF(A(I))102,101,102                                               PRT00390
  101 VLEN=X1(I)-XMIN                                                   PRT00400
      FLAG=1                                                            PRT00410
      RETURN                                                            PRT00420
  102 IF(X1(I).EQ.XMIN.AND.A(I).GT.0.) RETURN                           PRT00430
      Y=A(I)*XMIN+B(I)                                                  PRT00440
      IF(Y.LE.Y1(I).AND.Y.GE.Y2(I)) GO TO 105                           PRT00450
      GO TO 120                                                         PRT00460
  105 IF(Y.GE.YMIN.AND.Y.LE.YMAX) GO TO 110                             PRT00470
      GO TO 120                                                         PRT00480
  110 VLEN=SQRT((XMIN-X1(I))**2+(Y    -Y1(I))**2)                       PRT00490
      GO TO 440                                                         PRT00500
  120 Y=A(I)*XMAX+B(I)                                                  PRT00510
      IF(Y.LE.Y1(I).AND.Y.GE.Y2(I)) GO TO 125                           PRT00520
      GO TO 140                                                         PRT00530
```

```
  125 IF(Y .GE. YMIN .AND. Y .LE. YMAX) GO TO 150
      GO TO 140
  150 VLEN=SQRT((XMAX-X1(I))**2+(Y    -Y1(I))**2)
      GO TO 440
  140 X=(YMIN-B(I))/A(I)
      IF(A(I))141,142,142
  141 IF(X.GE.X1(I).AND.X.LE.X2(I)) GO TO 145
      GO TO 170
  142 IF(X.LE.X1(I).AND.X.GE.X2(I)) GO TO 145
      GO TO 170
  145 IF (X .GE. XMIN .AND. X .LE. XMAX) GO TO 165
      GO TO 170
  165 VLEN=SQRT((X    -X1(I))**2+(YMIN-Y1(I))**2)
      GO TO 440
  170 IF(A(I).GT.-573.) GO TO 175
      VLEN=Y1(I)-YMIN
      GO TO 440
  175 IF(A(I).LT.-.018.AND.A(I).GT.0.) GO TO 180
      VLEN=XMAX-X1(I)
      GO TO 440
  180 IF(A(I).LT.0..AND.A(I).GT..018) GO TO 185
      VLEN=X1(I)-XMIN
      GO TO 440
  185 WRITE(IPRINT,190) X1(I),Y1(I),X2(I),Y2(I)
      GO TO 422
C
C     CALCULATES VECLEN WHEN VECTOR END IS WITHIN GRID CELL
C
  200 IF(A(I))202,201,202
  201 VLEN=XMAX-X2(I)
      GO TO 440
  202 Y=A(I)*XMIN+B(I)
      IF(Y.LE.Y1(I).AND.Y.GE.Y2(I)) GO TO 205
      GO TO 220
  205 IF(Y .GE. YMIN .AND. Y .LE. YMAX) GO TO 210
      GO TO 220
  210 VLEN=SQRT((XMIN-X2(I))**2+(Y    -Y2(I))**2)
      GO TO 440
  220 Y=A(I)*XMAX+B(I)
      IF(Y.LE.Y1(I).AND.Y.GE.Y2(I)) GO TO 225
      GO TO 240
  225 IF(Y.GE.YMIN.AND.Y.LE.YMAX) GO TO 250
      GO TO 240
  250 VLEN=SQRT((XMAX-X2(I))**2+(Y    -Y2(I))**2)
      GO TO 440
  240 X=(YMAX-B(I))/A(I)
      IF(A(I))241,242,242
  241 IF(X.GE.X1(I).AND.X.LE.X2(I)) GO TO 245
      GO TO 270
  242 IF(X.LE.X1(I).AND.X.GE.X2(I)) GO TO 245
      GO TO 270
  245 IF(X.GE.XMIN.AND.X.LE.XMAX) GO TO 265
      GO TO 270
  265 VLEN=SQRT((X    -X2(I))**2+(YMAX-Y2(I))**2)
      GO TO 440
  270 IF(A(I).GT.-573.) GO TO 275
      VLEN=YMAX-Y2(I)
      GO TO 440
  275 IF(A(I).LT.-.018.AND.A(I).GT.0.) GO TO 280
      VLEN=X2(I)-XMIN
      GO TO 440
```

```
  280 IF(A(I).LT.0..AND.A(I).GT..018) GO TO 285
      VLEN=XMAX-X2(I)
      GO TO 440
  285 WRITE(IPRINT,290) X1(I),Y1(I),X2(I),Y2(I)
      GO TO 422
C
C     "PASS" DETERMINES WHETHER VECTOR PASSES THROUGH CELL
C     AND IF SO, ITS LENGTH WITHIN THE CELL
C
  300 FLAG=0
      NCLC1=0
      NCLC2=0
      NCLC3=0
      NCLC4=0
C     IF(Y1(I).LE.YMIN) RETURN
      IF(Y1(I).LT.YMIN) RETURN
      IF(Y2(I).GE.YMAX) RETURN
      IF(Y1(I).GE.YMIN.AND.Y1(I).LE.YMAX.AND.X1(I).LE.XMIN.AND.A(I).GT.0
     1.) RETURN
      IF(Y1(I).GE.YMIN.AND.Y1(I).LE.YMAX.AND.X1(I).GE.XMAX.AND.A(I).LT.0
     1.) RETURN
      IF(Y2(I).GE.YMIN.AND.Y2(I).LE.YMAX.AND.X2(I).LE.XMIN.AND.A(I).LT.0
     1.) RETURN
      IF(Y2(I).GE.YMIN.AND.Y2(I).LE.YMAX.AND.X2(I).GE.XMAX.AND.A(I).GT.0
     1.) RETURN
      IF(Y1(I).GE.YMAX.AND.X1(I).GE.XMAX.AND.A(I).LT.0.) RETURN
      IF(Y1(I).GE.YMAX.AND.X1(I).LE.XMIN.AND.A(I).GT.0.) RETURN
C     IF(Y2(I).LE.YMIN.AND.X2(I).LE.XMIN.AND.A(I).LT.0.) RETURN
      IF(Y2(I).LE.YMIN.AND.X2(I).GE.XMAX.AND.A(I).GT.0.) RETURN
      IF(Y2(I).LE.YMIN.AND.X2(I).LE.XMIN.AND.A(I).LT.0..AND.A(I).GT.
     1-573.) RETURN
      IF(A(I).EQ.0.0.AND.X1(I).GE.XMAX.AND.X2(I).LE.XMIN.AND.Y1(I).GE.YM
     1IN.AND.Y2(I).LT.YMAX) GO TO 301
      IF(A(I).EQ.0.0.AND.X2(I).GE.XMAX) RETURN
      IF(A(I).EQ.0.0.AND.X1(I).LE.XMIN) RETURN
      IF(A(I).GT.-573.) GO TO 303
      IF(X1(I).GE.XMIN.AND.X1(I).LT.XMAX.AND.X2(I).GE.XMIN.AND.X2(I).LT.
     1XMAX.AND.Y1(I).GE.YMAX.AND.Y2(I).LE.YMIN.AND.A(I).LE.-573.) GO TO
     1302
      RETURN
  301 VLEN=XMAX-XMIN
      GO TO 440
  302 VLEN=YMAX-YMIN
      GO TO 440
  303 YXMIN=A(I)*XMIN+B(I)
      IF(YXMIN.LE.Y1(I).AND.YXMIN.GE.Y2(I)) GO TO 305
      GO TO 320
  305 IF(YXMIN.GE.YMIN.AND.YXMIN.LE.YMAX) GO TO 310
      GO TO 320
  310 NCLC1=1
  320 YXMAX=A(I)*XMAX+B(I)
      IF(YXMAX.LE.Y1(I).AND.YXMAX.GE.Y2(I)) GO TO 325
      GO TO 340
  325 IF(YXMAX.GE.YMIN.AND.YXMAX.LE.YMAX)  GO TO 330
      GO TO 340
  330 NCLC2=1
      IF(NCLC1.GT.0.AND.NCLC2.GT.0) GO TO 335
      GO TO 340
  335 VLEN=SQRT((YXMAX-YXMIN)**2+(XMIN-XMAX)**2)
      GO TO 440
  340 XYMIN=(YMIN-B(I))/A(I)
```

```
      IF(A(I))341,342,342                                                PRT01760
  341 IF(XYMIN.GE.X1(I).AND.XYMIN.LE.X2(I)) GO TO 345                    PRT01770
      GO TO 380                                                          PRT01780
  342 IF(XYMIN.LE.X1(I).AND.XYMIN.GE.X2(I)) GO TO 345                    PRT01790
  345 IF(XYMIN.GE.XMIN.AND.XYMIN.LE.XMAX) GO TO 350                      PRT01800
      GO TO 380                                                          PRT01810
  350 NCLC3=1                                                            PRT01820
      IF(NCLC2.GT.0.AND.NCLC3.GT.0) GO TO 360                            PRT01830
      GO TO 365                                                          PRT01840
  360 VLEN=SQRT((XMAX-XYMIN)**2+(YMIN-YXMAX)**2)                         PRT01850
      GO TO 440                                                          PRT01860
  365 IF(NCLC1.GT.0.AND.NCLC3.GT.0) GO TO 370                            PRT01870
      GO TO 380                                                          PRT01880
  370 VLEN=SQRT((XMIN-XYMIN)**2+(YXMIN-YMIN)**2)                         PRT01890
      GO TO 440                                                          PRT01900
  380 XYMAX=(YMAX-B(I))/A(I)                                             PRT01910
      IF(A(I))381,382,382                                                PRT01920
  381 IF(XYMAX.GE.X1(I).AND.XYMAX.LE.X2(I)) GO TO 385                    PRT01930
      GO TO 422                                                          PRT01940
  382 IF(XYMAX.LE.X1(I).AND.XYMAX.GE.X2(I)) GO TO 385                    PRT01950
      GO TO 422                                                          PRT01960
  385 IF(XYMAX.GE.XMIN.AND.XYMAX.LE.XMAX) GO TO 390                      PRT01970
      FLAG=0                                                             PRT01980
      RETURN                                                             PRT01990
  390 NCLC4=1                                                            PRT02000
      IF(NCLC3.GT.0.AND.NCLC4.GT.0) GO TO 395                            PRT02010
      GO TO 400                                                          PRT02020
  395 VLEN=SQRT((XYMAX-XYMIN)**2+(YMAX-YMIN)**2)                         PRT02030
      GO TO 440                                                          PRT02040
  400 IF(NCLC2.GT.0.AND.NCLC4.GT.0) GO TO 405                            PRT02050
      GO TO 410                                                          PRT02060
  405 VLEN=SQRT((XMAX-XYMAX)**2+(YMAX-YXMAX)**2)                         PRT02070
      GO TO 440                                                          PRT02080
  410 IF(NCLC1.GT.0.AND.NCLC4.GT.0) GO TO 425                            PRT02090
      IF(NCLC1.GE.1.OR.NCLC2.GE.1.OR.NCLC3.GE.1.OR.NCLC4.GE.1) GO TO 420PRT02100
  420 WRITE(IPRINT,430) NCLC1,NCLC2,NCLC3,NCLC4,X1(I),Y1(I),X2(I),Y2(I) PRT02110
  422 FLAG=0                                                             PRT02120
      RETURN                                                             PRT02130
  425 VLEN=SQRT((XYMAX-XMIN)**2+(YMAX-YXMIN)**2)                         PRT02140
  440 FLAG=1                                                             PRT02150
      IF(VLEN.LT.MINLEN) FLAG=0                                          PRT02160
      RETURN                                                             PRT02170
  190 FORMAT(1H0,70H********ERROR********** VECTOR ORIGIN SUBROUTINE. VECPRT02180
     1TOR DELIMITERS ARE,4(5X,F7.1),//,1H ,50H**ERROR MESSAGE REFERS TO PRT02190
     2FOLLOWING PRINTED CELL**)                                          PRT02200
  290 FORMAT(1H0,67H********ERROR********* VECTOR END SUBROUTINE. VECTORPRT02210
     1 DELIMITERS ARE,4(5X,F7.1),//,1H ,50H**ERROR MESSAGE REFERS TO FOLPRT02220
     2LOWING PRINTED CELL**)                                             PRT02230
  430 FORMAT(1H0,63H******ERROR****** SUBROUTINE VECTOR-PASS; VALUES OF PRT02240
     1NCLC1-4 ARE,4I5  ,22H VECTOR DELIMITERS ARE,4F7.1)                 PRT02250
      END                                                                PRT02260
      SUBROUTINE MID                                                     MID00010
C                                                                        MID00020
C     SUBROUTINE 'MID' DETERMINES IF VECTOR MIDPOINT FALLS WITHIN GRID  MID00030
C     CELL                                                               MID00040
C                                                                        MID00050
      COMMON X1(2000),Y1(2000),X2(2000),Y2(2000),VECAZM(2000),VEC_EN(200MID00060
     1O),A(2000),B(2000),XMID(2000),YMID(2000),CLAMIN(90),CLAMAX(90),   MID00070
     2AZLEN(90),NAZFRQ(90),XMIN,YMIN,XMAX,YMAX,FLAG,VLEN,I               MID00080
      INTEGER XMIN,YMIN,XMAX,YMAX,FLAG                                   MID00090
      IF(XMID(I).GE.XMIN.AND.XMID(I).LT.XMAX.AND.YMID(I).GE.YMIN.AND.YMIMID00100
```

```
     1D(I).LT.YMAX) GO TO 1
      FLAG=0
      RETURN
    1 FLAG=-1
      RETURN
      END
      FUNCTION PRBCHI (CHISQ,IDF)
C
C     WRITTEN BY H.D. KNOBLE & F.YATES BORDEN, THE PENNSYLVANIA STATE
C     UNIVERSITY, 1966
C     THIS FUNCTION COMPUTES BY THE APPROXIMATIONS ON PAGE 941 OF
C     "HANDBOOK OF MATHEMATICAL FUNCTIONS",U.S. DEPT. OF COMMERCE,1964.
C     GIVEN A VALUE OF CHI-SQUARE AND ITS DEGREES OF FREEDOM, FUNCTION
C     PRBCHI COMPUTES THE PROBABILITY OF A GREATER VALUE OF CHI-SQUARE.
C     THE Z(ARGUMENT) FUNCTION IS COMPUTED BY FORMULA 26.2.1, P. 931.
C
      INTEGER TEST
C     ALL REAL*8 ARGUMENTS CHANGED TO DOUBLE PRECISION BY M.PODWYSOCKI.
      DOUBLE PRECISION DSQRT,DEXP,ARG,SCHISQ,XPLEVL
      DOUBLE PRECISION Q,R,S,T,U,V,V9,PROB,S2PI,Z005,APPROX
      DATA S2PI/2.5066282D00/
C
      Q(ARG)=(DEXP(-ARG*ARG*0.5)/2.5066282D00)*(T*(0.3193815D00+T*
     1(-0.3565638D00+T*(1.781478D00+T*(-1.821256D00+(1.330274D00*T))))))
      XPL=2.5762359600
      PRBCHI=0.0
      IF(CHISQ.LT.0.0)  RETURN
      IF(IDF.LE.0) RETURN
  100 SCHISQ=CHISQ
      S=1.0
      V=IDF
      V9=2.0/FLOAT(9*IDF)
      U=-SCHISQ*0.5
      SCHISQ=DSQRT(SCHISQ)
      IF (DABS(U).LT.174.6) GO TO 110
C
C     174.6 IS THE LARGEST ARGUMENT THAT EXP WILL TAKE.
C
      PROB=0.0
      GO TO 240
C
C   CHECK FOR DEGREES OF FREEDOM GREATER THAN 100 OR GREATER THAN 30
C
  110 IF (IDF.GT.100) GO TO 200
      IF (IDF.GT.30) GO TO 170
C
C   DEGREES OF FREEDOM LESS THAN OR EQUAL TO 30
C
      PROB=0.0
      TEST=MOD(IDF,2)
      IF (TEST.NE.0) GO TO 140
C
C   EVEN DEGREES OF FREEDOM  **  LESS THAN OR EQUAL TO 30  ** FORMULA
C   26.4.5, PAGE 941
C
      IRANGE=(IDF-2)/2
      IF (IRANGE.EQ.0) GO TO 130
      DO 120  I=1,IRANGE
      IR=I+I
      S=S*IR
  120 PROB=PROB+SCHISQ**IR/S
```

```
  130 PROB=DEXP(U)*(1.0+PROB)
      GO TO 230
C
C   ODD DEGREES OF FREEDOM  **  LESS THAN OR EQUAL TO 29  ** FORMULA
C   26.4.4, PAGE 941
C
  140 IRANGE=(IDF-1)/2
      IF (IRANGE.EQ.0) GO TO 160
      DO 150  I=1,IRANGE
      IR=I+I-1
      S=S*IR
  150 PROB=PROB+SCHISQ**IR/S
  160 T=1.0/(1.0+0.2316419D00*SCHISQ)
      PROB=2.0*(Q(SCHISQ))+2.0*(DEXP(U)/S2PI)*PROB
      GO TO 230
C
C   ********** GREATER THAN 30 DEGREES OF FREEDOM **********
C   AN APPROXIMATE VALUE OF CHISQ IS FIRST COMPUTED THEN COMPARED WITH
C   THE GIVEN CHISQ. IF THE APPROX. VALUE IS GREATER THAN THE GIVEN
C   VALUE, Q(CHISQ,IDF) IS RETURNED AS .995.
C   ***********************************************************************
C   FOR GREATER THAN 30 AND LESS THAN OR EQUAL TO 100 DEGREES OF FREEDOM
C   THE APPROX. VALUE OF CHISQ AT THE .995 LEVEL IS COMPUTED BY FORMULA
C   26.4.17, PAGE 941.  THE SIGN OF X(P) IN THE FORMULA WAS CHANGED
C   FROM + TO - TO ALLOW COMPUTATION OF CHISQ AT THE .995 LEVEL RATHER
C   THAN THE .005 LEVEL AS IS THE CASE WHEN THE SIGN IS +.
C
  170 APROX=((1.0-V9-XPL*DSQRT(V9))**3)*V
      IF (APROX.LE.CHISQ) GO TO 180
      GO TO 210
  180 V=((CHISQ/V)**0.3333333333D00-(1.0-V9))/DSQRT(V9)
  190 T=1.0/(1.0+0.2316419D00*V)
      PROB=Q(V)
      GO TO 230
C
C   GREATER THAN 100 DEGREES OF FREEDOM.    THE APPROX. VALUE OF CHISQ
C   IS COMPUTED BY FORMULA 26.4.16, PAGE 941.  THE SIGN OF X(P) WAS
C   CHANGED FOR THE SAME REASON AS ABOVE.
C
  200 APROX=((-XPL+DSQRT(V+V-1.0))**2)*0.5
C
      IF (APROX.LE.CHISQ) GO TO 220
  210 PROB=+0.995
      GO TO 240
  220 V=DSQRT(2.0D0*CHISQ)-DSQRT(2.0*V-1.0)
      GO TO 190
  230 IF (PROB.GT.0.995) GO TO 210
  240 PRBCHI=PROB
      RETURN
      END
```

ROSE DIAGRAM PROGRAM

PRECEDING PAGE BLANK NOT FILMED

```
C              **********************************                       ROSE0010
C                   ROSE   DIAGRAM   PROGRAM                            ROSE0020
C              **********************************                       ROSE0030
C                                                                       ROSE0040
C       THE PROGRAM WAS WRITTEN BY MELVIN H. PODWYSOCKI OF THE GEO-     ROSE0050
C       SCIENCES DEPT., THE PENNSYLVANIA STATE UNIVERSITY, MAY,1972 FORROSE0060
C       USE ON AN IBM 360/67 COMPUTER ALONG WITH THE PENN STATE COMPU-  ROSE0070
C       TATION CENTER "ODGS" GRAPHICS PACKAGE. THE PROGRAM WAS MODIFIEDROSE0080
C       IN MAY,1973 TO USE STANDARD CALCOMP SOFTWARE SUBROUTINES WITH   ROSE0090
C       THE CALCOMP 780 DRUM PLOTTER PACKAGE ON THE IBM 360 SERIES      ROSE0100
C       COMPUTERS HAVING THE EQUIVALENT OF 130K BYTES STORAGE.           ROSE0110
C                                                                       ROSE0120
C     THE PROGRAM PRODUCES ROSE DIAGRAMS SUITABLE FOR MAP OVERLAY FOR   ROSE0130
C     ANY NUMBER OF MAP GRID CELLS, PLOTTING THE ROSE AT THE MIDPOINT   ROSE0140
C     OF THE CELL ALONG WITH THE SUM TOTAL OF THE VALUES COMPRISING     ROSE0150
C     THE ROSE.  SCALING FACTORS ARE INCLUDED WITHIN THE PROGRAM.       ROSE0160
C     DATA INPUT IS GENERATED BY THE "AZMAP" PROGRAM AND CONSISTS OF X ROSE0170
C     AND Y MIDPOINTS (XMID & YMID) OF EACH GRID CELL AND THE COMPO-    ROSE0180
C     NENTS OF THE ROSE DIAGRAM (AZLEN).  UP TO 90 AZIMUTH CLASSES MAY ROSE0190
C     BE USED BETWEEN 270 THRU 0 TO 90 DEGREES.  COORDINATES MUST BE    ROSE0200
C     READ IN MM., X IS + TO THE RIGHT AND Y IS + DOWNWARD.  NORTH IS   ROSE0210
C     ASSUMED PARALLEL TO THE Y AXIS.  CONTROL CARDS ARE READ FROM THE ROSE0220
C     CARD READER WHILE DATA CARDS ARE READ FROM ANY UNIT DECLARED BY   ROSE0230
C     'ITAPE1' IN STATEMENT ROS 660.   OUTPUT IS GENERATED AS PER THE   ROSE0240
C     CALCOMP PACKAGE AT EACH INDIVIDUAL INSTALLATION AND CONSISTS OF   ROSE0250
C     A PLOT OF X & Y AXES AND ROSE DIAGRAMS.  THE TOTAL VALUE OF UNITSROSE0260
C     COMPRISING EACH ROSE DIAGRAM IS ALSO PLOTTED.                     ROSE0270
C                                                                       ROSE0280
C     ALL NUMERIC INPUT DATA IS RIGHT JUSTIFIED: "I" INDICATES INTEGER,ROSE0290
C     "F" INDICATES FLOATING POINT AND "A" INDICATES CHARACTER FORMAT; ROSE0300
C     "#" PRECEEDING NUMBERS INDICATES COLUMNS USED FOR EACH PARAMETER.ROSE0310
C                                                                       ROSE0320
C ********CONTROL CARD 1-------PARAMETER CARD                           ROSE0330
C    XMN=MINIMUM X VALUE FOR PLOT IN MM. (F7.2,#1-7)                    ROSE0340
C    YMN=MINIMUM Y VALUE FOR PLOT IN MM. (F7.2,#8-14)                   ROSE0350
C    XMX=MAXIMUM X VALUE FOR PLOT IN MM. (F7.2,#15-21)                  ROSE0360
C    YMX=MAXIMUM Y VALUE FOR PLOT IN MM. (F7.2,#22-28)                  ROSE0370
C       NOTE: THE ABOVE 4 VALUES GOVERN THE X AND Y AXES LABELING,      ROSE0380
C       WHICH WILL BE INCREMENTED BY THE VALUE 'SC' (ROS 650).  SO      ROSE0390
C       THAT WHEN 'FACT' = 1, 1 INCH = 'SC' MM.                         ROSE0400
C    FACT=MULTIPLICATION FACTOR FOR ALL CARTESIAN COORDINATE DATA; AL- ROSE0410
C       LOWS SCALING LARGE DIMENSIONED MAPS TO SIZE ACCOMMODATED BY     ROSE0420
C       PLOTTER.  MAY BE <, = OR > 1, BUT NOT < 0. (F7.2,#29-35)        ROSE0430
C    RSSZ=MULTIPLICATION FACTOR FOR 'AZLEN' OF ROSE DIAGRAM.  SCALES    ROSE0440
C       ROSE DIAGRAMS SO THAT THEY DO NOT OVERLAP EACH OTHER OR EXCEED ROSE0450
C       BOUNDS OF PLOT AND IS EMPERICALLY DEVELOPED. (F7.2,#36-42)      ROSE0460
C       NOTE: ALL 'AZLEN' VALUES ARE SCALED BY 'FACT' AS WELL AS 'SC'   ROSE0470
C    NCELL=NUMBER OF MAP GRID CELLS (I4,#43-46)                         ROSE0480
C    NCLASS=NUMBER OF AZIMUTH CLASSES (I.E. 180 DEGREES/10 DEGREE CLASSROSE0490
C       INCREMENT = 18 CLASSES)  (I2,#47-48)                            ROSE0500
C    PGS=Y AXIS PAGESIZE OF CALCOMP PLOTTER (IN INCHES, I.E. 12.,30.,   ROSE0510
C       ETC.)  (F5.0,#49-53)                                            ROSE0520
C ********CONTROL CARD 2-------TITLE CARD                               ROSE0530
C    TITLE WILL BE PLACED AT TOP OF PLOTTED OUTPUT.  IF CENTERING IS    ROSE0540
C       DESIRED, IT MUST BE PUNCHED SYMMETRICALLY ABOUT #40 OF THE      ROSE0550
C       TITLE CARD (20A4,#1-80)                                         ROSE0560
C ********CONTROL CARD 3-------INPUT FORMAT CARD                        ROSE0570
C    FORMAT MUST BE ENCLOSED IN PARANTHESES AND BEGIN IN #1.  SEQUENCE ROSE0580
C       MUST BE: XMID, YMID, AND THE NUMBER OF 'AZLEN' CLASSES AS SPE- ROSE0590
C       CIFIED BY 'NCLASS' (I.E. (2F5.2,8F6.2/10F6.2) FOR 'NCLASS' =    ROSE0600
```

PRECEDING PAGE BLANK NOT FILMED

```
C         18)   (20A4,1-80)
C ********DATA CARDS------
C      READ FROM ANY UNIT DECLARED BY 'ITAPE1'.  MUST BE IN FORMAT SPECI-
C         FIED BY  CONTROL CARD 3.
C
      DIMENSION AZLEN(180),AZMID(180),AZX(180),AZY(180),ABX(180),
     1ABY(180),AZRAD(180)
      DIMENSION BUFFER (8000),FMTRD(20),TITLE(20)
      DATA PRAD/1.745329E-2/,IREAD/5/,SC/50./,HT25/.25/,HT1/.1/,
     1ITAPE1/5/
      CALL PLOTS(BUFFER,8000)
C
C     READ CONTROL CARDS
C
      READ(IREAD,5) XMN,YMN,XMX,YMX,FACT,RSSZ,NCELL,NCLASS,PGS
      READ(IREAD,10) (TITLE(L),L=1,20)
      READ(IREAD,10) (FMTRD(L),L=1,20)
C
C     MOVE PEN TO TOP OF PAGE (ASSUMES PEN SET TO PAGE BOTTOM)
C
      PGSZ=PGS-0.15*PGS
      CALL PLOT(0.,PGSZ,23)
C
C     FACTOR THE FOLLOWING PLOTTING SUBROUTINES
C
      CALL FACTOR(FACT)
C
C     PLOT X AXIS PARALLEL TO TOP OF PAGE
C
      CALL AXIS(0.,0.,13HX AXIS OF MAP,13,(XMX-XMN)/SC,0.,XMN,50.,10.)
C
C     PLOT NORTH ARROW PARALLEL TO Y AXIS (SIDE OF PAGE)
C
      ANRTX=XMX/SC+1.0
      ANRT=XMX/SC+1.5
      CALL PLOT(ANRTX,-3.,3)
      CALL PLOT(ANRTX,-1.5,2)
      CALL PLOT(ANRT,-2.,2)
      CALL SYMBOL(ANRTX-HT25/4,-1.25,HT25,1HN,0.,1)
C
C     PLOT TITLE ABOVE X AXIS
C
      TSTRT=((XMX-XMN)/(SC*2.))-0.5*80.*HT25
      CALL SYMBOL(TSTRT,1.25,HT25,TITLE,0.,80)
C
C     PLOT Y AXIS PARALLEL TO SIDE OF PAGE
C
      CALL AXIS(0.,0.,13HY AXIS OF MAP,-13,(YMX-YMN)/SC,-90.,YMN,50.,10.
     1)
      NC=2*NCLASS
C
C     READ DATA FROM UNIT ITAPE1, ONE ROSE DIAGRAM AT A TIME
C
      DO 200 K=1,NCELL
      READ(ITAPE1,FMTRD) XMID,YMID,(AZLEN(M),M=1,NCLASS)
      TOTLN=0.
C
C     PLOT EACH ROSE DIAGRAM
C
      DO 140 I=1,NC
      NC1=NCLASS+I
```

```
      IF(I-NCLASS) 15,15,100
   15 IF(I-1) 40,40,60
   40 YSCALE=-YMID/SC
      XMID=XMID/SC
C
C     PLOT CENTERPOINT OF EACH ROSE DIAGRAM
C
      CALL SYMBOL(XMID-HT1/4.,YSCALE-HT1/2.,HT25*RSSZ,1H+,0.,1)
   60 IF(I-NCLASS) 80,80,100
   80 AZMID(I)=270.-180./NC+I*180./NCLASS
      IF(AZMID(I).GT.360.) AZMID(I)=AZMID(I)-360.
      AZMID(NC1)=AZMID(I)+180.
      IF(AZMID(NC1).GT.360.) AZMID(NC1)=AZMID(NC1)-360.
      AZLEN(NC1)=AZLEN(I)
      TOTLN=TOTLN+AZLEN(I)
  100 IF(AZMID(I).GE.0..AND.AZMID(I).LT.90.) AZMID(I)=90.-AZMID(I)
      IF(AZMID(I).GE.90..AND.AZMID(I).LT.360.) AZMID(I)=450.-AZMID(I)
      AZRAD(I)=AZMID(I)*PRAD
C
C     SCALE AZLEN BY RSSZ AND SC
C
      ABX(I)=(AZLEN(I)*COS(AZRAD(I))*RSSZ)/SC
      AZX(I)=ABX(I)+XMID
      ABY(I)=(AZLEN(I)*SIN(AZRAD(I))*RSSZ)/SC
      AZY(I)=ABY(I)+YSCALE
C
C     PLOT EACH AZLEN
C
      KEY=2
      IF(I.EQ.1) KEY=3
      CALL PLOT(AZX(I),AZY(I),KEY)
      IF(I-NC) 140,120,120
  120 CALL PLOT(AZX(1),AZY(1),2)
  140 CONTINUE
      STRT=AZX(NCLASS)+0.5*RSSZ
C
C     PLOT AZLEN SUM FOR EACH ROSE DIAGRAM
C
      CALL NUMBER(STRT,AZY(NCLASS),2.*HT25*RSSZ,TOTLN,0.,1)
  200 CONTINUE
C
C     TERMINATE PLOT
C
      CALL PLOT(XMX/SC+5.,(YMN-YMX)/SC,999)
    5 FORMAT(6F7.2,I4,I2,F5.0)
   10 FORMAT(20A4)
      STOP
      END
```

NASA–GSFC